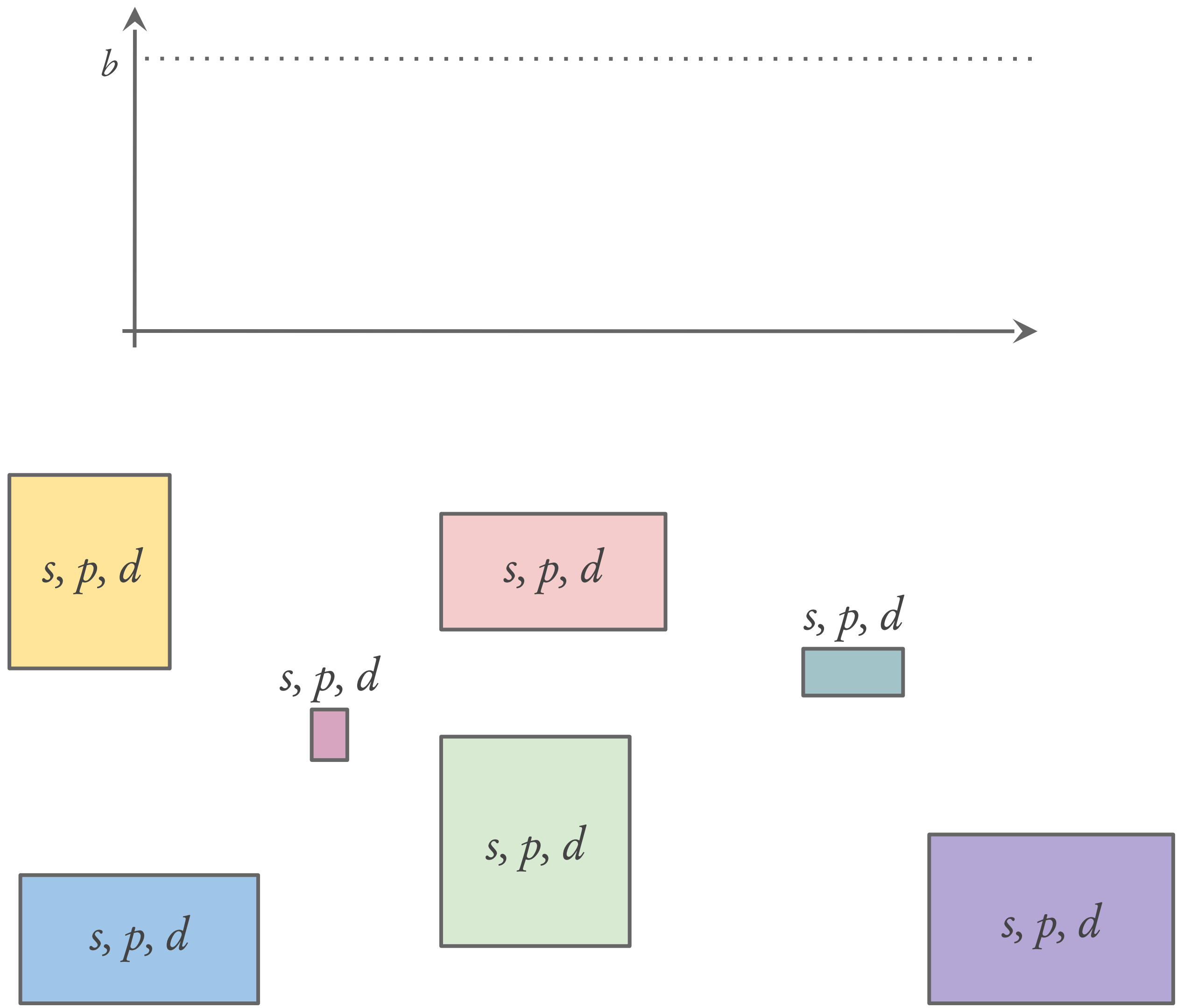
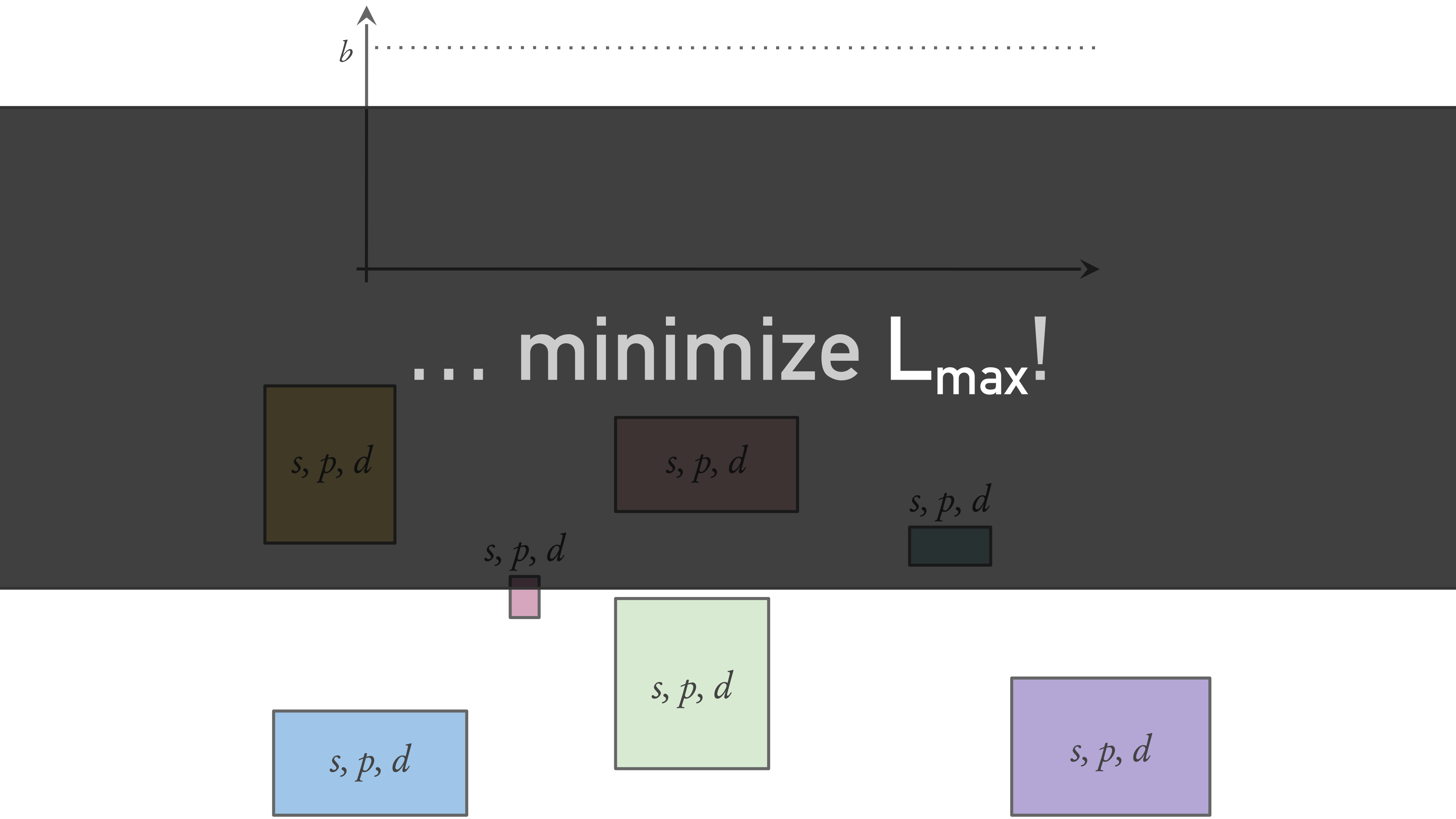
Scheduling non-identical jobs on a batch machine

Sebastian Kosch
BASc Thesis, EngSci 1T3
Feb 27th, 2013

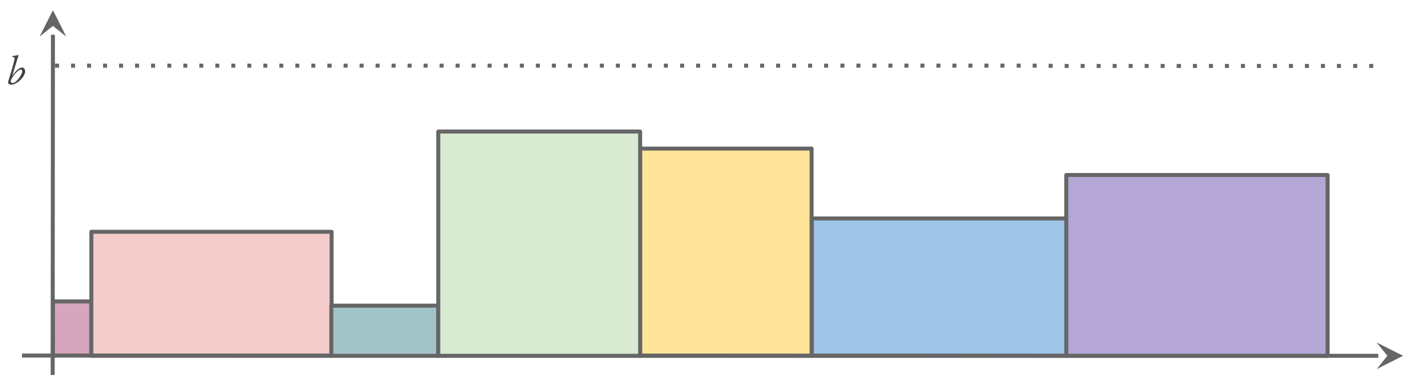


b
s, p, d
s, p, d
s, p, d
s, p, d
s, p, d
s, p, d
s, p, d

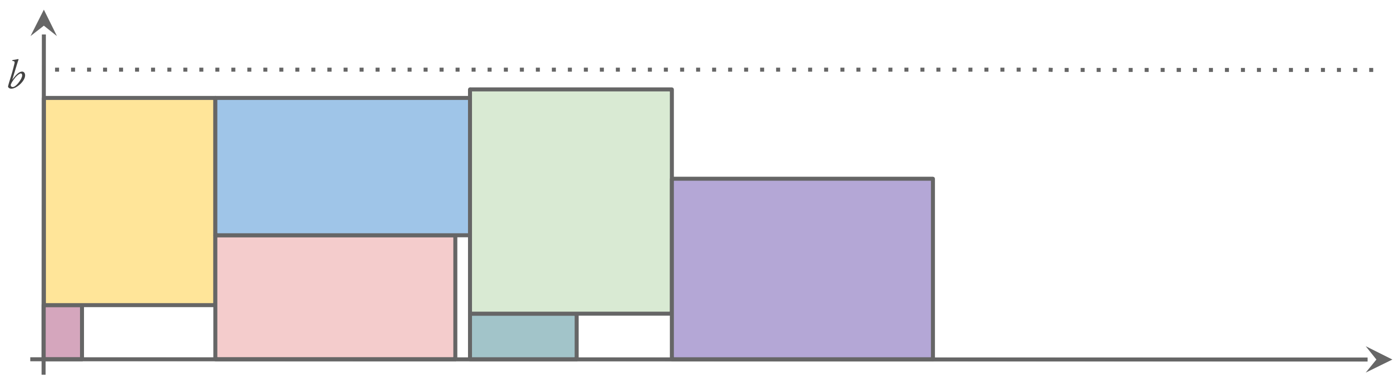
Scheduling non-identical jobs on a batch machine: **Intro**



… minimize L_{max}!



Earliest Due Date First (EDD)

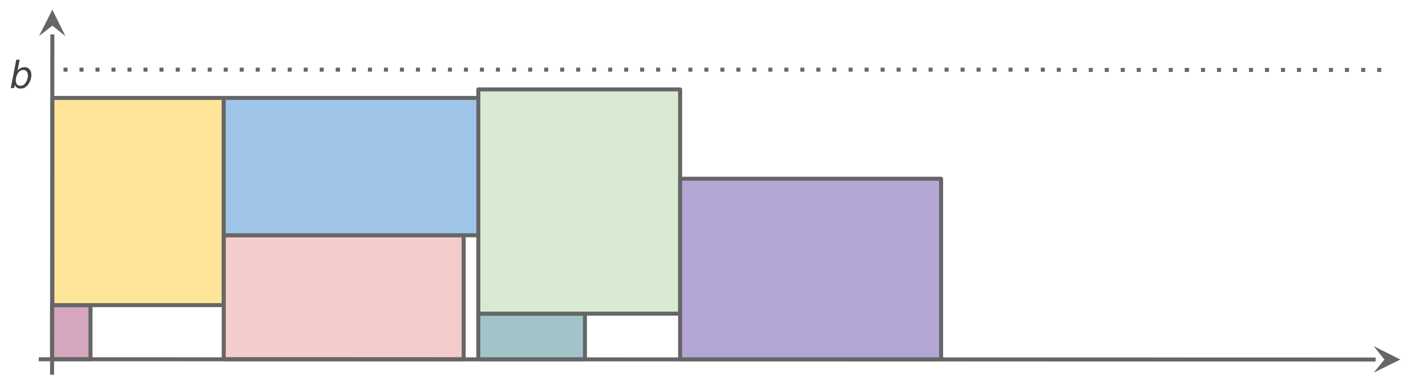


Jobs batched

Batches due with earliest-due job

Earliest Due Date First (EDD)


HELLER 733



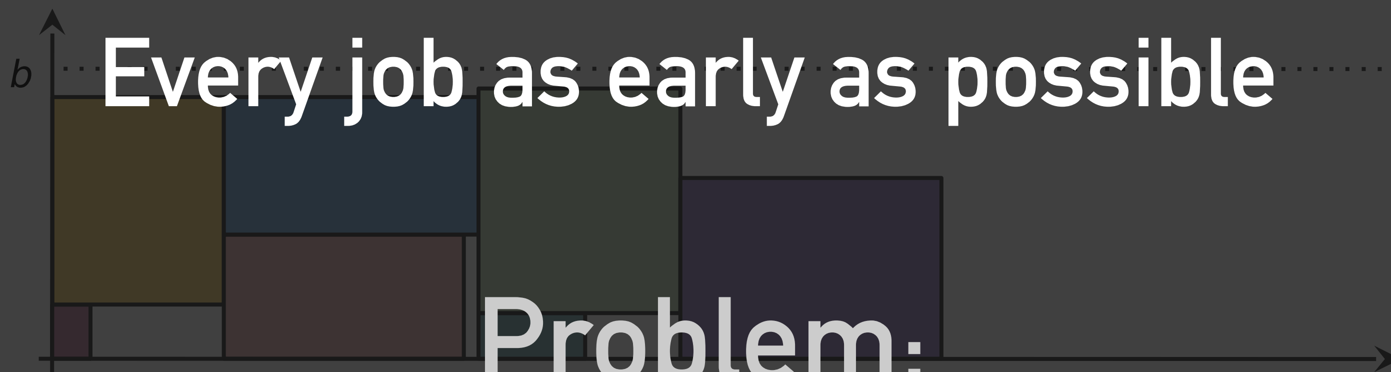
Jobs batched

Batches due with earliest-due job

Earliest Due Date First (EDD)

Objective:

Every job as early as possible



Problem:

Position of L_{max} is unknown, so which jobs should be batched?

Batches due with earliest-due job

Earliest Due Date First (EDD)

European Journal of Operational Research 221 (2012) 533–545

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

A constraint programming approach for a batch processing problem with non-identical job sizes

Arnaud Malapert [a,c,*], Christelle Guéret [b], Louis-Martin Rousseau [c]

[a] École des Mines de Nantes, LINA UMR CNRS 6241, Nantes, France
[b] École des Mines de Nantes, IRCCyN UMR CNRS 6597, Nantes, France
[c] École Polytechnique de Montréal, CIRRELT, Montréal, Québec, Canada

ARTICLE INFO

Article history:
Received 28 April 2011
Accepted 6 April 2012
Available online 17 April 2012

Keywords:
Combinatorial optimization
Artificial intelligence
Constraint programming
Scheduling
Packing

ABSTRACT

This paper presents a constraint programming approach for a batch processing machine on which a finite number of jobs of non-identical sizes must be scheduled. A parallel batch processing machine can process several jobs simultaneously and the objective is to minimize the maximal lateness. The constraint programming formulation proposed relies on the decomposition of the problem into finding an assignment of the jobs to the batches, and then minimizing the lateness of the batches on a single machine. This formulation is enhanced by a new optimization constraint which is based on a relaxed problem and applies cost-based domain filtering techniques. Experimental results demonstrate the efficiency of cost-based domain filtering techniques. Comparisons to other exact approaches clearly show the benefits of the proposed approach: it can optimally solve problems that are one order of magnitude greater than those solved by a mathematical formulation or by a branch-and-price.

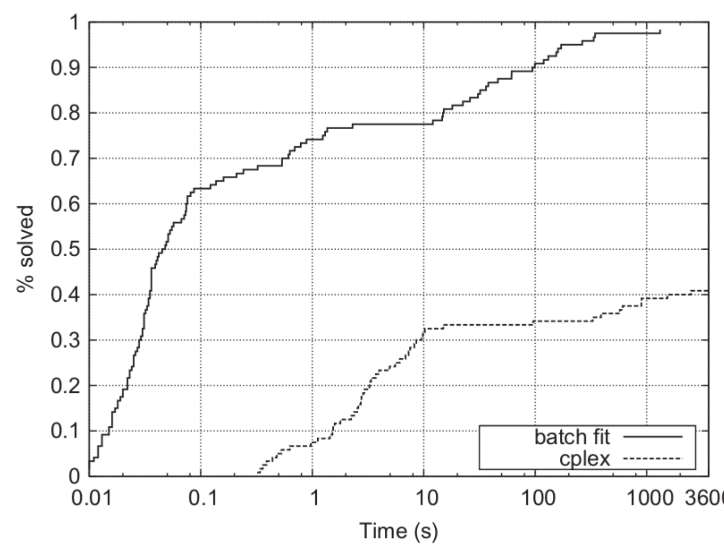
1. Introduction

This paper presents a constraint programming approach for a batch processing machine on which a finite number of jobs with non-identical sizes must be scheduled. A parallel batch processing machine can process several jobs simultaneously. Such machines

namic job arrivals and the objective of minimizing the makespan. Since then, heuristics (Perez et al., 2000; Wang and Uzsoy, 2002; Uzsoy, 1995), genetic algorithm (Wang and Uzsoy, 2002) and exact methods (Webster and Baker, 1995; Mehta and Uzsoy, 1998; Liu et al., 2007) have been proposed for identical job sizes and due date related performance measures. Several papers describe approaches

Malapert's `sequenceEDD` global constraint

- Update bounds on $L_{\max}$ based on known P_k and D_k in partial assignment
- Eliminate possible assignments based on known bounds on $L_{\max}$
- Update bounds on n_k based on minimum cost of opening a new batch

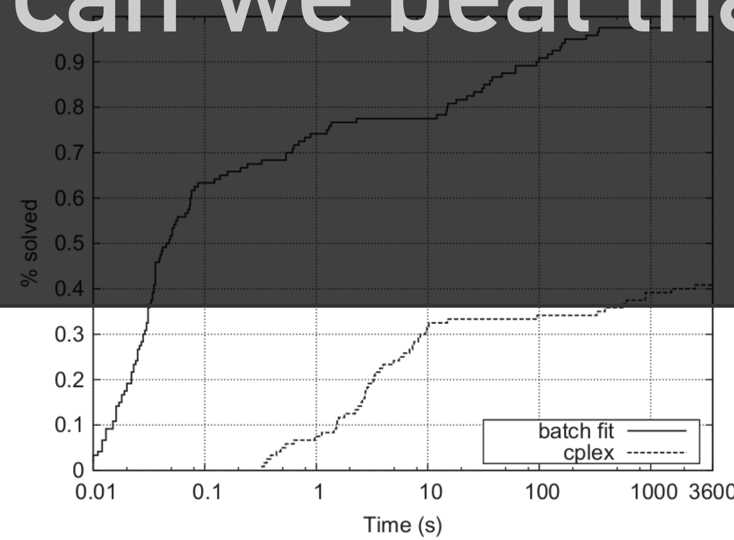


(a) Time comparison ($n \leq 50$).

Malapert's `sequenceEDD` global constraint

- Update bounds on $L_{\max}$ based on known P_k and D_k in partial assignment
- Eliminate possible assignments based on known bounds ond $L_{\max}$
- Update bounds on n_k based on minimum cost of opening a new batch

… can we beat that?



(a) Time comparison ($n \leq 50$).

Scheduling non-identical jobs on a batch machine: MIP approach I

$$
\begin{array}{lll}
\text{Min.} & L_{\max} & \\
\text{s.t.} & \displaystyle\sum_{k \in K} x_{jk} = 1 & \forall j \in J \\
& \displaystyle\sum_{j \in J} s_j x_{jk} \le b & \forall k \in K \\
& p_j x_{jk} \le P_k & \forall j \in J, \forall k \in K \\
& C_{k-1} + P_k = C_k & \forall k \in K \\
& (d_{max} - d_j)(1 - x_{jk}) + d_j \ge D_k & \forall j \in J, \forall k \in K \\
& D_{k-1} \le D_k & \forall k \in K \\
& C_k - D_k \le L_{\max} & \forall k \in K \\
& C_k \ge 0, P_k \ge 0 \text{ and } D_k \ge 0 & \forall k \in K
\end{array}
$$

Scheduling non-identical jobs on a batch machine: **MIP approach I**

group empty batches

$$\sum_{j \in J} x_{j,k-1} = 0 \rightarrow \sum_{j \in J} x_{jk} = 0 \quad \forall k \in K$$

$$e_k + \sum_{j \in J} x_{jk} \geq 1 \quad \forall k \in K$$

$$n_j(e_k - 1) + \sum_{j \in J} x_{jk} \leq 0 \quad \forall k \in K$$

$$e_k - e_{k-1} \geq 0 \quad \forall k \in K$$

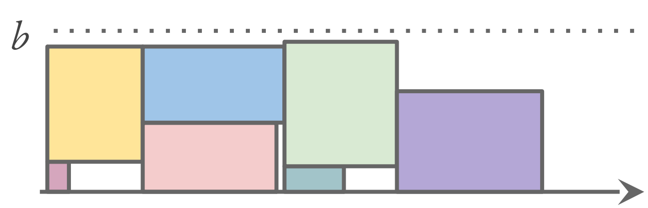
only move jobs back

$$x_{jk} = 0 \quad \forall \{j \in J, k \in K | j > k\}$$

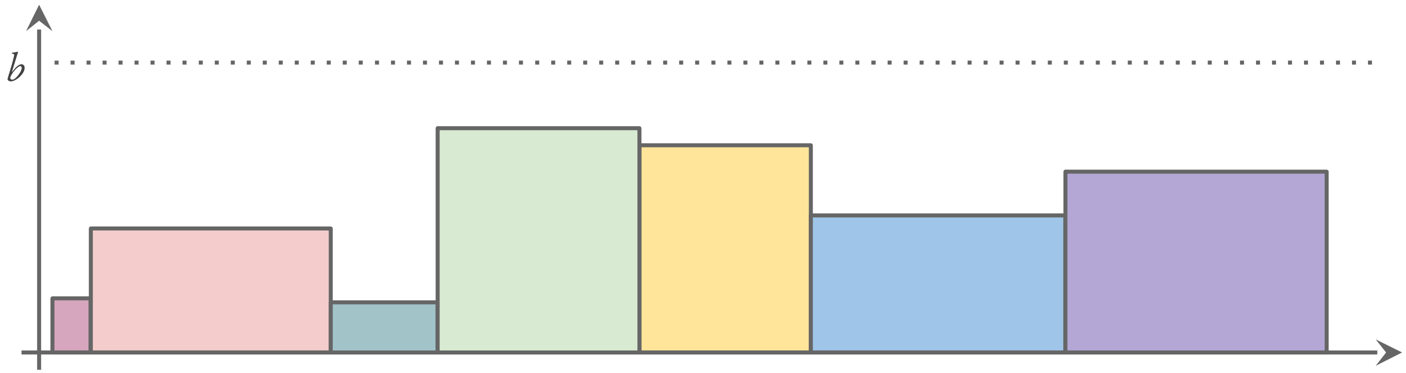
bound on L_{max}

$$L_{\max} \geq \lceil \frac{1}{b} \sum_{j} s_j p_j \rceil - \delta_q \quad \forall q, \forall \{j \in J | d_j \leq \delta_q\}$$

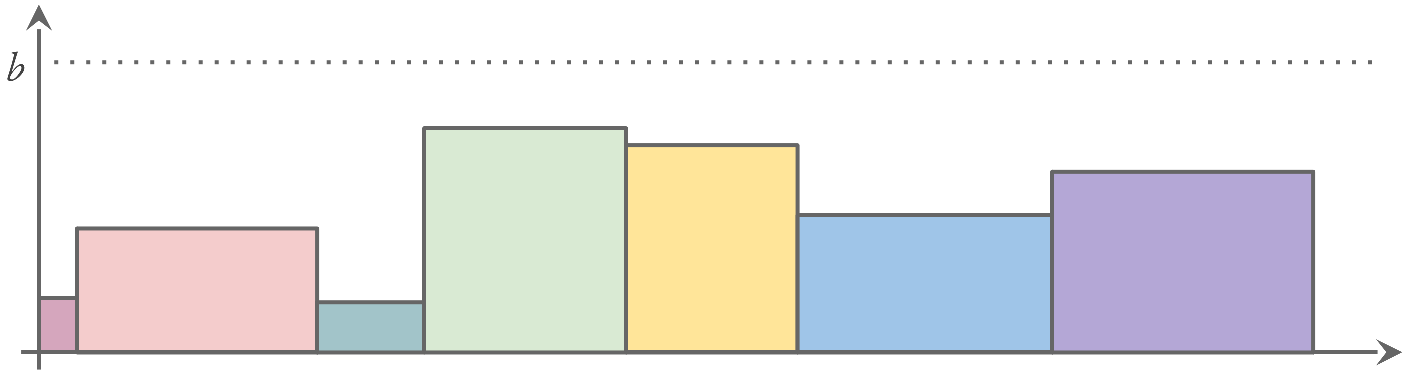
$$
\begin{array}{lll}
\text{Min.} & L_{\max} & \\
\text{s.t.} & \texttt{pack}(J, K, b) & \\
 & \texttt{cumul}(J, b) & \\
 & P_k = \max_j p_j & \forall \{j \in J \mid B_j = k\}, \forall k \in K \\
 & D_k = \min_j d_j & \forall \{j \in J \mid B_j = k\}, \forall k \in K \\
 & C_k + P_{k+1} = C_{k+1} & \forall k \in K \\
 & L_{\max} \geq \max_k (C_k - D_k) & \\
 & \texttt{IfThen}(P_k = 0, P_{k+1} = 0) & \forall k \in \{k_1, \ldots, k_{n_k - 2}\} \\
 & B_j \leq k & \forall \{j \in J, k \in K \mid j > k\}
\end{array}
$$

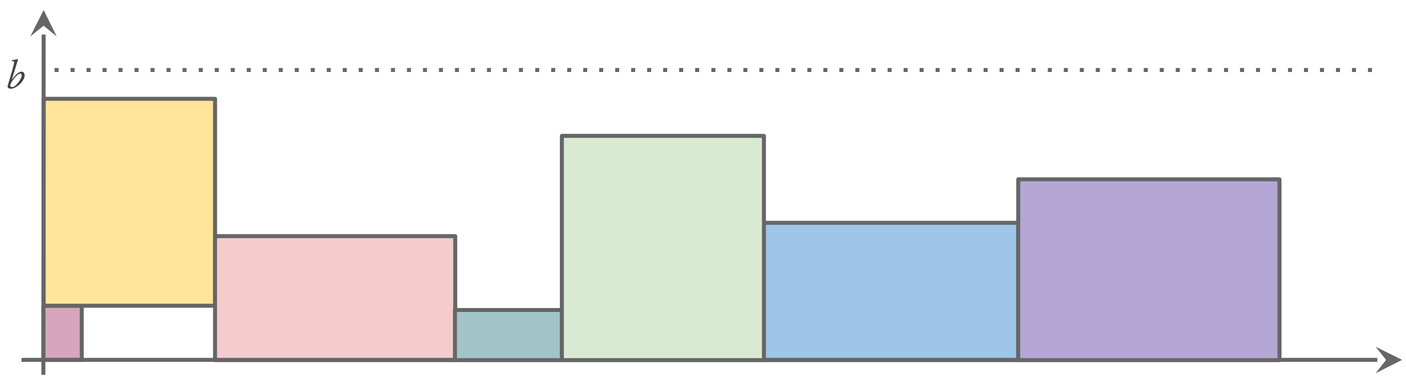


Scheduling non-identical jobs on a batch machine: **batch-by-batch approach**

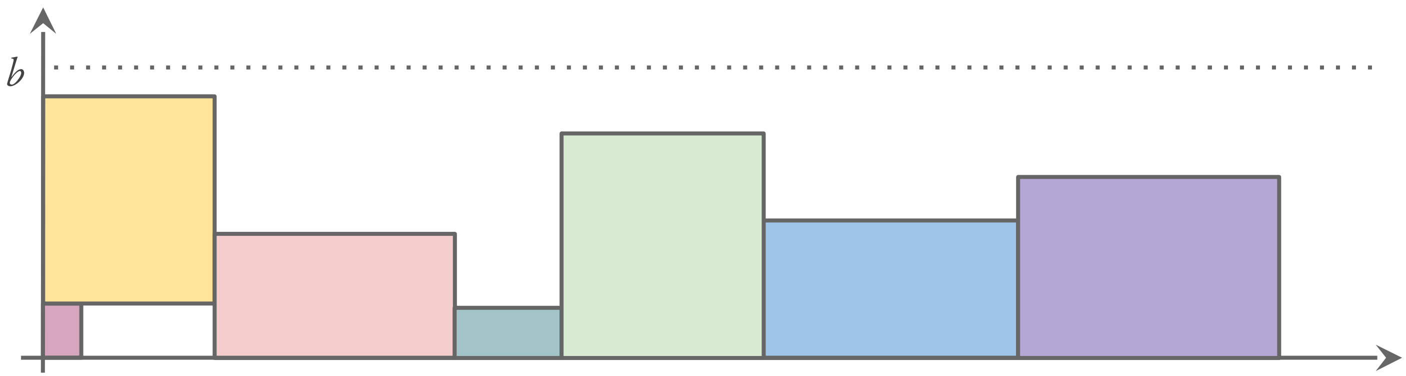


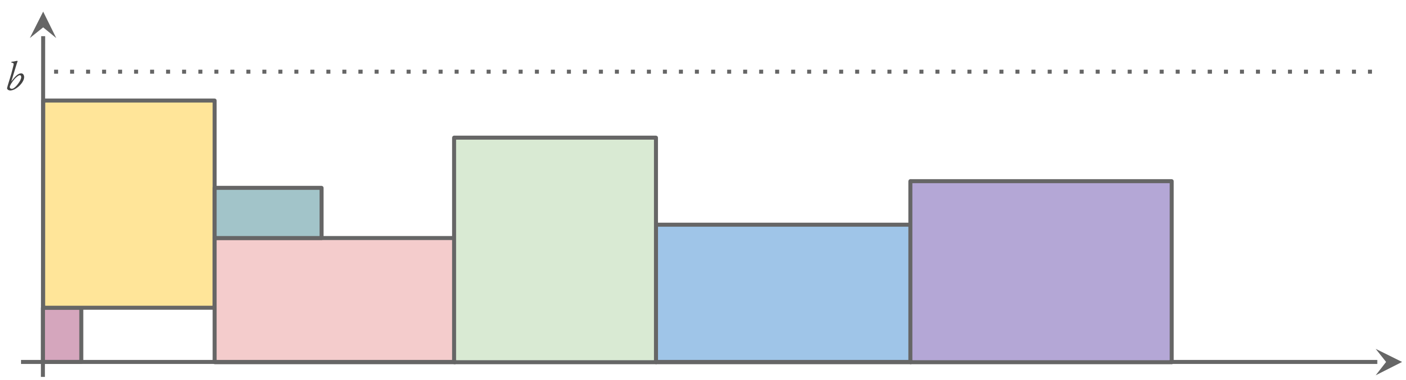
Scheduling non-identical jobs on a batch machine: **batch-by-batch approach**



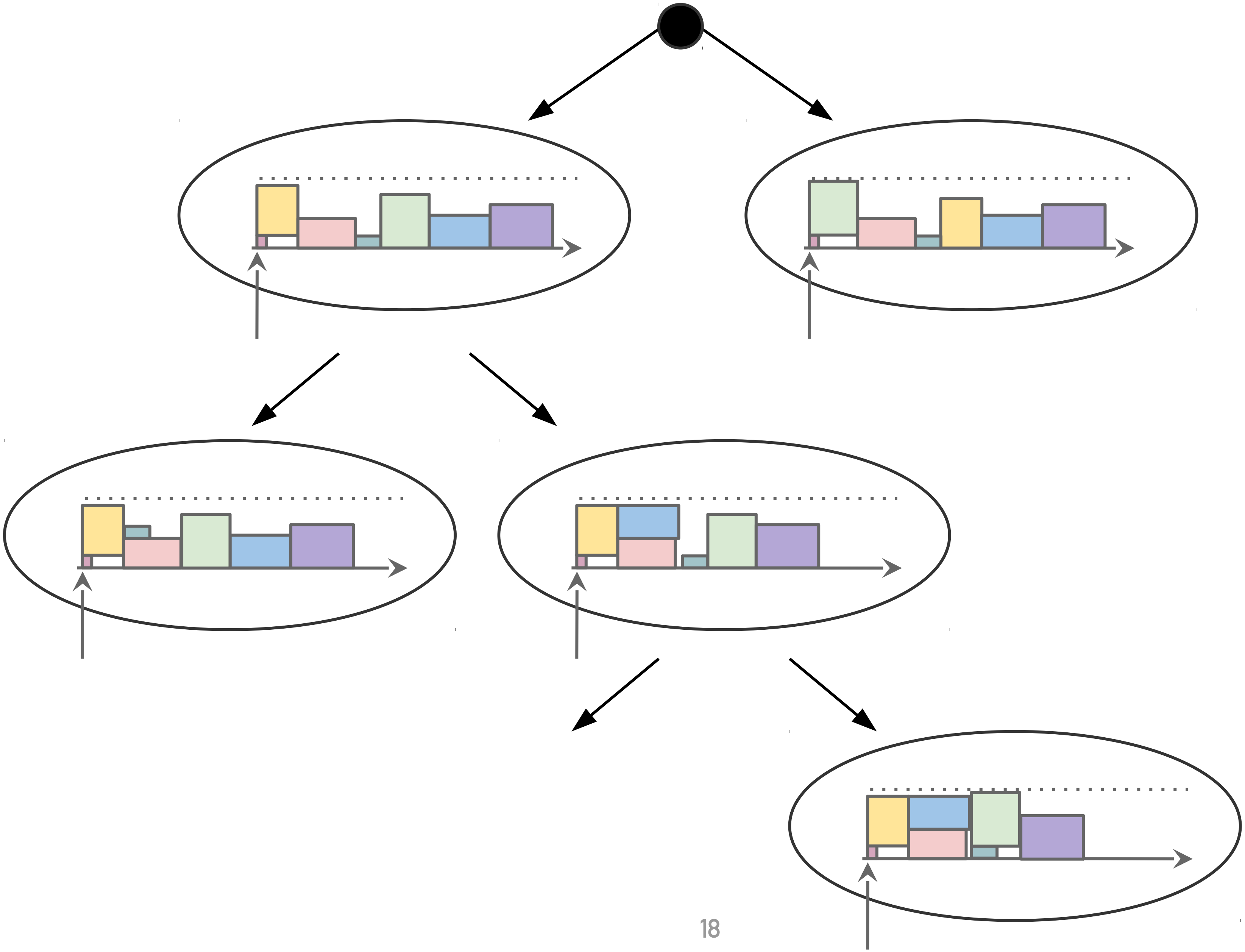


Scheduling non-identical jobs on a batch machine: **batch-by-batch approach**





Scheduling non-identical jobs on a batch machine: **batch-by-batch approach**

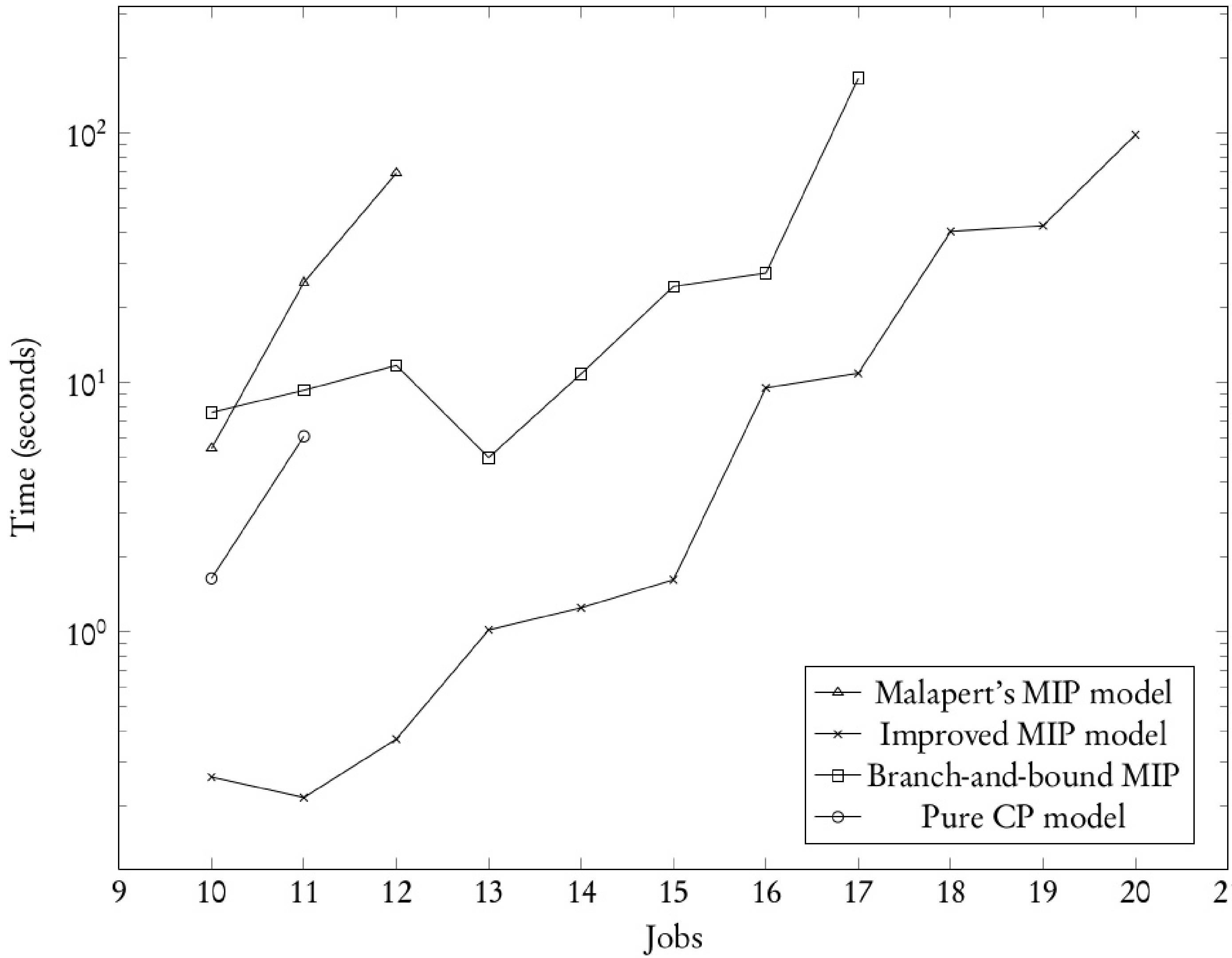


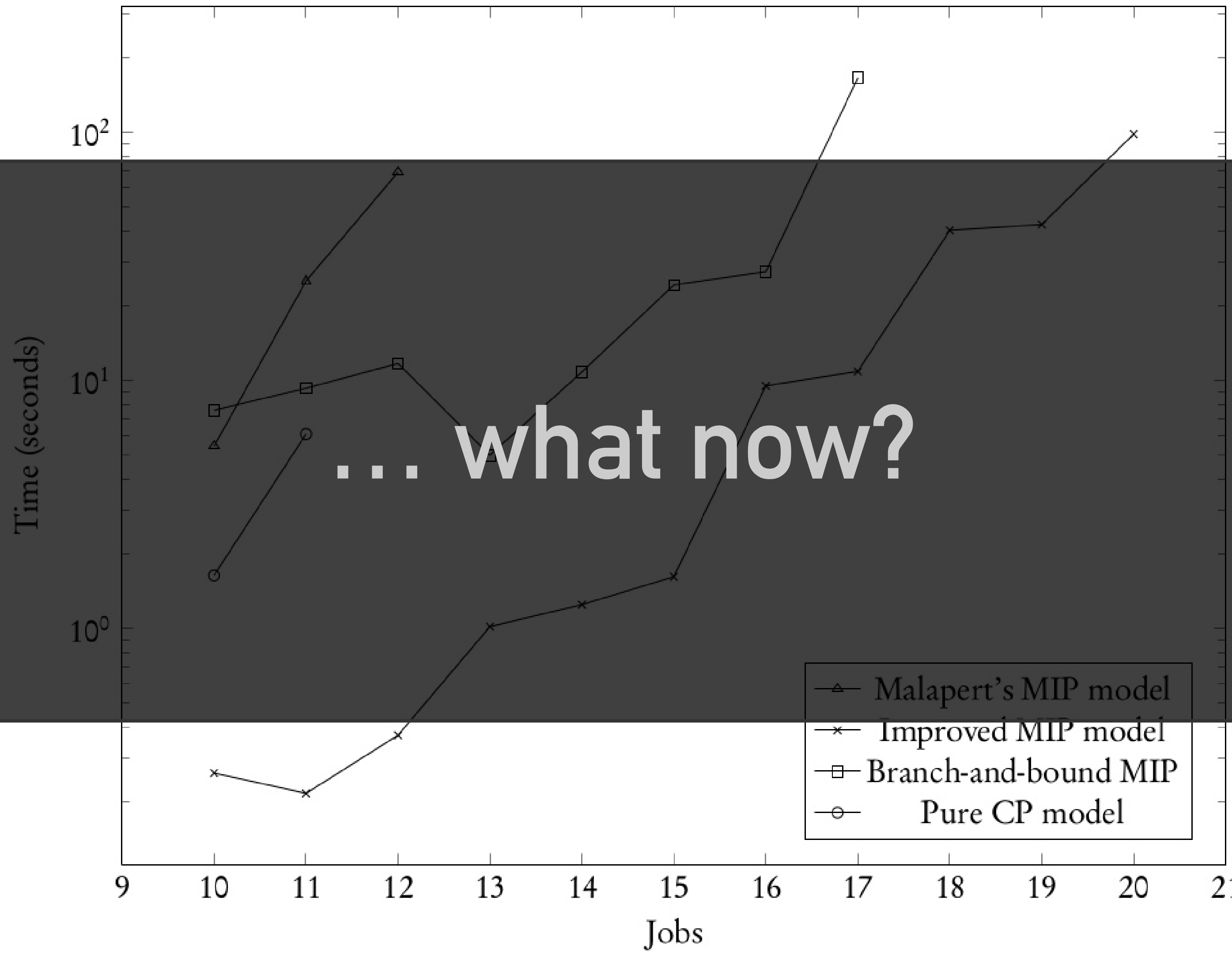
$$
\begin{array}{lll}
\text{Min.} & L_{\text{max,cumul}} & \\
\text{s. t.} & \sum_j s_j x_j \leq b & \forall j \in J \\
& P_k \geq p_j x_j & \forall j \in J \\
& \sum_j x_j \geq 1 & \forall \{j \in J | d_j = \min(d_j)\} \\
& P_k + \frac{1}{b} \sum_i s_i p_i \leq d_j + L_{\text{max,incmb}} - 1 - v_k & \forall j \in J, \forall \{i \in J | d_i \leq d_j\} \\
& \sum_t u_{jt} = 1 & \forall j \in J \\
& \sum_j \sum_{t' \in T_{jt}} s_j u_{jt'} \leq b & \forall t \in \mathcal{H} \\
& (v_k + t + p_j) u_{jt} \leq d_j + L_{\text{max,incmb}} - 1 & \forall j \in J, \forall t \in \mathcal{H} \\
& L_{\text{max,cumul}} \geq (v_k + t + p_j) u_{jt} - d_j & \forall j \in J, \forall t \in \mathcal{H} \\
& u_{j,t=0} = x_j & \forall j \in J \\
& u_{it} \leq (1 - x_j) & \forall i, j \in J, \forall t \in \{1, \ldots, p_j - 1\} \\
& b - \sum_{i \in J} s_i x_i \leq (b w_j + 1) s_j & \forall j \in J \\
& P_k + 2 w_j n_t \geq p_j + n_t x_j & \forall j \in J \\
& P_k - 2(1 - w_j) n_t \leq p_j + n_t x_j - 1 & \forall j \in J
\end{array}
$$

$$
\begin{aligned}
\text{Min.} \quad & L_{\max} \\
\text{s.t.} \quad & \texttt{startOf}(j) \geq (1 - x_j) P_k && \forall j \in J \\
& \texttt{startOf}(j) \leq (1 - x_j) \sum_{i \in J} p_i && \forall j \in J \\
& x_j \geq \frac{\frac{1}{2} - \texttt{startOf}(j)}{\sum_{i \in J} p_i} && \forall j \in J \\
& x_j \leq 1 + \frac{\frac{1}{2} - \texttt{startOf}(j)}{\sum_{i \in J} p_i} && \forall j \in J \\
& P_k \geq \max_{j \in J} (x_j p_j) \\
& \texttt{endOf}(j) = d_j + L_{\max,\text{incmb}} - 1 && \forall j \in J \\
& L_{\max} \geq \max_{j \in J} (\texttt{endOf}(j) - d_j) \\
& \texttt{IfThen}(p_j \leq P_k \wedge x_j = 0, b - \sum_{j \in J} s_j x_j \leq s_j) && \forall j \in J \\
& P_k + \frac{1}{b} \sum_i s_i p_i \leq d_j + L_{\max,\text{incmb}} - 1 - v_k && \forall j \in J, \forall \{i \in J \mid d_i \leq d_j\} \\
& \sum_j x_j \geq 1 && \forall \{j \in J \mid d_j = \min(d_j)\} \\
& \texttt{cumul}(J, b)
\end{aligned}
$$

$$
\begin{aligned}
\text{Min.} \quad & L_{\max} \\
\text{s.t.} \quad & \texttt{startOf}(j) \geq (1 - x_j) P_k && \forall j \in J \\
& \texttt{startOf}(j) \leq (1 - x_j) \sum_{i \in J} p_i && \forall j \in J \\
& x_j \geq \frac{\frac{1}{2} - \texttt{startOf}(j)}{\sum_{i \in J} p_i} && \forall j \in J \\
& x_j \leq 1 + \frac{\frac{1}{2} - \texttt{startOf}(j)}{\sum_{i \in J} p_i} && \forall j \in J \\
& P_k \geq \max_{j \in J} (x_j p_j) && \\
& \texttt{endOf}(j) = d_j + L_{\max,\text{incmb}} - 1 && \forall j \in J \\
& L_{\max} \geq \max_{j \in J} (\texttt{endOf}(j) - d_j) && \\
& \texttt{IfThen}(p_j \leq P_k \wedge x_j = 0, b - \sum_{j \in J} s_j x_j \leq s_j) && \forall j \in J \\
& P_k + \frac{1}{b} \sum_i s_i p_i \leq d_j + L_{\max,\text{incmb}} - 1 - v_k && \forall j \in J, \forall \{i \in J \mid d_i \leq d_j\} \\
& \sum_j x_j \geq 1 && \forall \{j \in J \mid d_j = \min(d_j)\} \\
& \texttt{cumul}(J, b) &&
\end{aligned}
$$

Scheduling non-identical jobs on a batch machine: **Results**

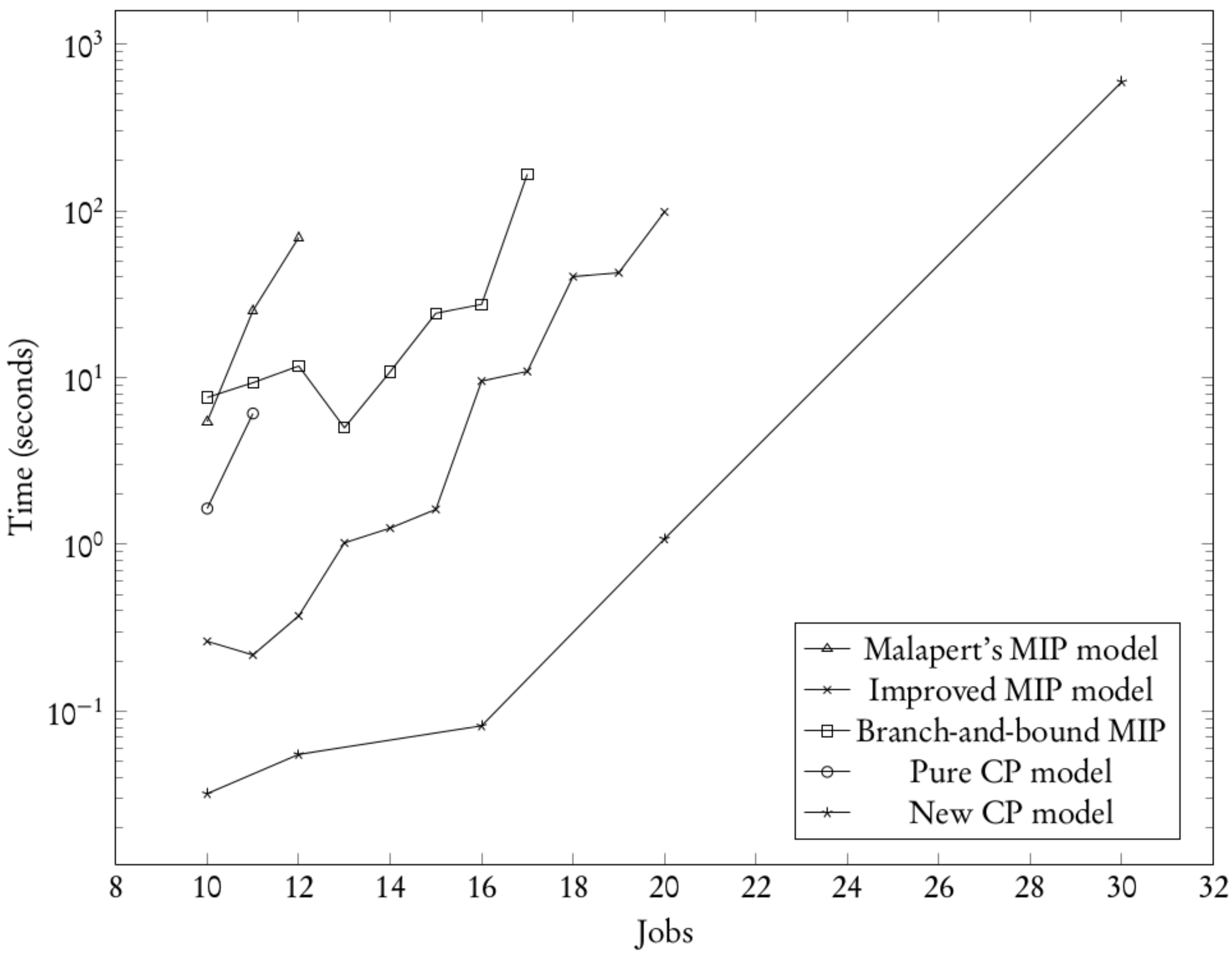


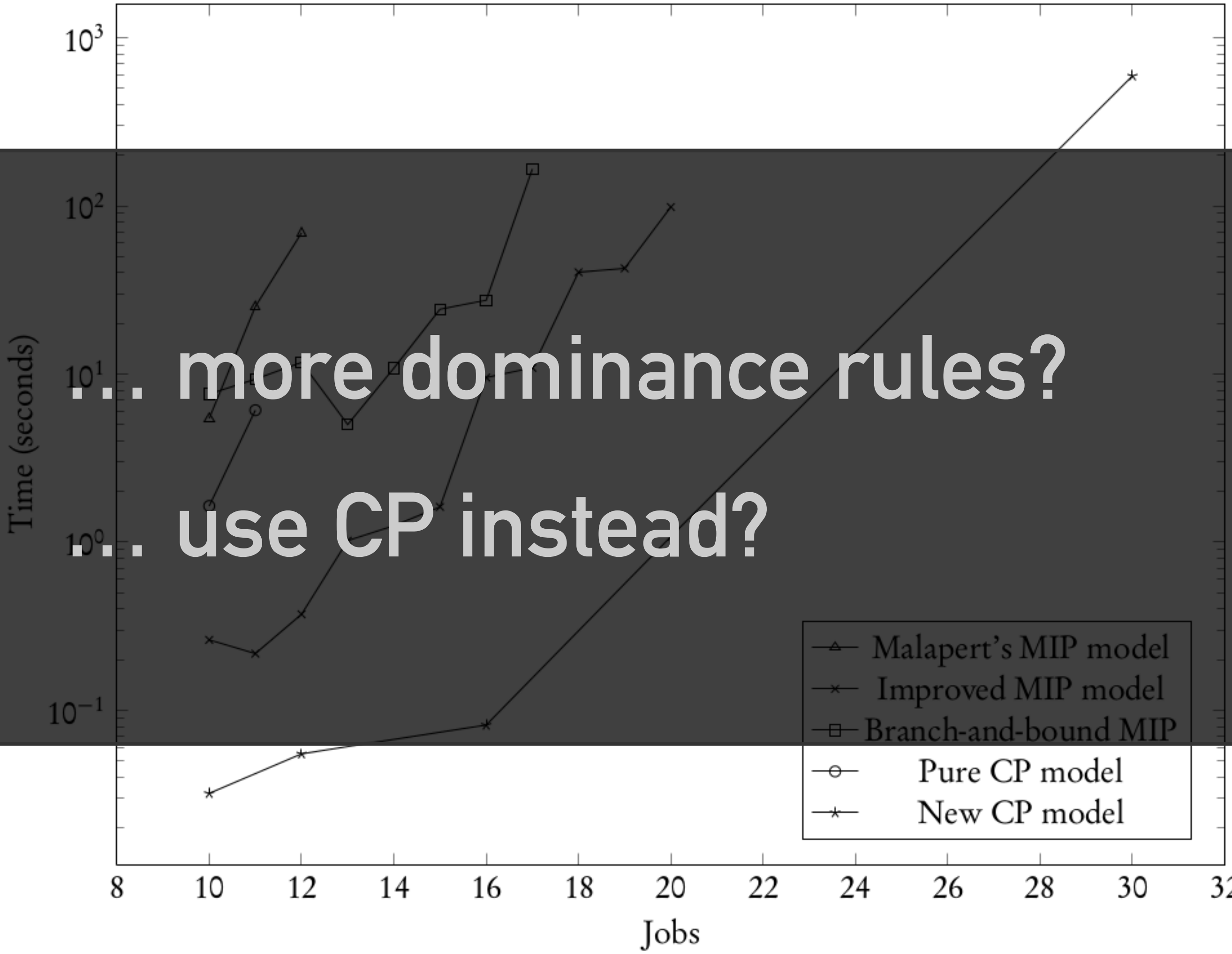

Time (seconds)
10^2
10^1
10^0
9
10
11
12
13
14
15
16
17
18
19
20
21
Jobs
Malapert's MIP model
Improved MIP model
Branch-and-bound MIP
Pure CP model

… what now?

$$
\begin{aligned}
\text{Min.}\quad & L_{\max} \\
\text{s.t.}\quad & \sum_{k \in K} x_{jk} = 1 && \forall j \in J \\
& \sum_{j \in J} s_j x_{jk} \le b && \forall k \in K \\
& x_{jk} = 0 && \forall \{j \in J, k \in K \mid j \le k \vee s_j + s_k > b\} \\
& L_{\max} \ge \ell_k + \sum_{h=0}^{k-1} \left[P_h - p_h \left(1 + \sum_{i \in K} x_{hi} \right) \right] - \mathrm{UB}(L_{\max}) \sum_{i \in K} x_{ki} && \forall k \in K \\
& p_j x_{jk} \le P_k && \forall j \in J, \forall k \in K
\end{aligned}
$$

Scheduling non-identical jobs on a batch machine: **Current work**





… more dominance rules?

… use CP instead?

Questions?